On Monotone and Convex Spline Interpolation*

By Paolo Costantini

Abstract. This paper is concerned with the problem of existence of monotone and/or convex splines, having degree n and order of continuity k, which interpolate to a set of data at the knots. The interpolating splines are obtained by using Bernstein polynomials of suitable continuous piecewise linear functions; they satisfy the inequality $k \leq n - k$. The theorems presented here are useful in developing algorithms for the construction of shape-preserving splines interpolating arbitrary sets of data points. Earlier results of McAllister, Passow and Roulier can be deduced from those given in this paper.

1. Introduction. Let $x_0 < x_1 < \cdots < x_N$ be real numbers, let n and k, $0 \leq k \leq n - 1$, be integers, and let

$$S_n^k[x_0, x_N] = \left\{ s \in C^k[x_0, x_N]: s(x) \in \mathbf{P}_n, \, x \in [x_i, x_{i+1}], \, i = 0, 1, \dots, N - 1 \right\}$$

be the set of splines of degree n and deficiency $n - k$ at knots x_i, $i = 1, \dots, N - 1$. In a previous paper [2], a necessary and sufficient condition for the existence of monotone and convex $s \in S_3^1[x_0, x_N]$ interpolating a set of data at the knots was given, and in [3] an algorithm for the construction of shape-preserving interpolating splines was described. Further investigations led to an extension of the above results to $s \in S_n^k[x_0, x_N]$, where $1 \leq k \leq n - k$ and $s^{(j)}(x_i) = 0$, $j = 2, \dots, k$, $i = 0, 1, \dots, N$, whenever $k \geq 2$. Such splines are obtained from Bernstein polynomials of monotone and/or convex interpolating linear splines with two knots in (x_i, x_{i+1}), $i = 0, \dots, N - 1$. The well-known theorems of [9] and [7] (see also [8] for related results), where linear splines with one knot in (x_i, x_{i+1}), $i = 0, \dots, N - 1$, are used, can be deduced from those presented here.

This paper is divided into three parts. In Section 2, n and k are prescribed and a data-dependent condition is given. In Section 3 the degree is computed to ensure the existence of a monotone and/or convex interpolating spline, and a degree-dependent condition is provided. Section 4 is then devoted to final conclusions and remarks.

The results of Sections 2 and 3 enable us to derive an algorithm for the construction of shape-preserving interpolating $s \in S_n^k[x_0, x_N]$. The spline can be selected by the user to perform Lagrange or Hermite interpolation, to be piecewise monotone and/or convex (concave), n and k can be assigned or automatically computed by the code, according to Theorem 3. A detailed description of this algorithm, together with numerical and graphical examples, will be reported elsewhere [1].

Finally, we note that all the results of this paper, which are stated for increasing (I), convex (C) or increasing and convex (IC) data, can be easily modified to the other cases.

2. Data-Dependent Existence Conditions. Let (x_i, y_i), $i = 0, 1, \ldots, N$, be a set of data points. Setting $h_i = x_{i+1} - x_i$, $\Delta_i = (y_{i+1} - y_i)/h_i$, $i = 0, \ldots, N - 1$, we say that the data are increasing if $\Delta_i \geqslant 0$, $i = 0, \ldots, N - 1$, and convex if $\Delta_{i+1} \geqslant \Delta_i$, $i = 0, \ldots, N - 2$. Now let the integers k, n_i, $1 \leqslant k \leqslant n_i - k$, $i = 0, \ldots, N - 1$, be given and let $n = \max\{n_i, i = 0, \ldots, N - 1\}$. We introduce the sets

$$(2.1) \quad \begin{aligned} LS_n^k[x_0, x_N] = \Big\{ s \in S_n^k[x_0, x_N]: s(x) \in \mathbf{P}_{n_i} \text{ for } x \in [x_i, x_{i+1}], \\ i = 0, \ldots, N - 1; \\ s^{(j)}(x_i^+) = 0, \ j = 2, \ldots, k, \ i = 0, \ldots, N - 1; \\ s^{(j)}(x_i^-) = 0, \ j = 2, \ldots, n_i - k, \ i = 1, \ldots, N \Big\}, \end{aligned}$$

$$(2.2) \quad \begin{aligned} RS_n^k[x_0, x_N] = \Big\{ s \in S_n^k[x_0, x_N]: s(x) \in \mathbf{P}_{n_i} \text{ for } x \in [x_i, x_{i+1}], \\ i = 0, \ldots, N - 1; \\ s^{(j)}(x_i^+) = 0, \ j = 2, \ldots, n_i - k, \ i = 0, \ldots, N - 1; \\ s^{(j)}(x_i^-) = 0, \ j = 2, \ldots, k, \ i = 1, \ldots, N \Big\}, \end{aligned}$$

$$(2.3) \quad \begin{aligned} BS_n^k[x_0, x_N] = RS_n^k[x_0, x_N] + LS_n^k[x_0, x_N] \\ = \Big\{ s \in S_n^k[x_0, x_N]: s = \phi + \psi, \ \phi \in LS_n^k[x_0, x_N], \\ \psi \in RS_n^k[x_0, x_N] \Big\}. \end{aligned}$$

For $k = 1$ or $n_i - k = 1$ the corresponding conditions are vacuous and we have $BS_2^1[x_0, x_N] \equiv S_2^1[x_0, x_N]$ and $BS_3^1[x_0, x_N] \equiv S_3^1[x_0, x_N]$ if $n_i = 3$, $i = 0, \ldots, N - 1$. We remark that $s^{(j)}(x_i) = 0$, $j = 2, \ldots, k$, $i = 0, 1, \ldots, N$, for any $s \in BS_n^k[x_0, x_N]$. We need the set of points t_i, $i = 0, 1, \ldots, 2N + 1$, where

$$(2.4) \quad \begin{aligned} t_0 = x_0; \qquad t_{2N+1} = x_N; \\ t_{2i+1} = x_i + \frac{k}{n_i} h_i; \quad t_{2i+2} = x_i + \frac{n_i - k}{n_i} h_i, \qquad i = 0, \ldots, N - 1, \end{aligned}$$

which we use to define

$$(2.5) \quad \begin{aligned} LS_1^0[t_0, t_{2N+1}] = \big\{ l \in C[t_0, t_{2N+1}]: l(x) \in \mathbf{P}_1 \text{ for } x \in [t_0, t_1]; \\ x \in [t_{2i+1}, t_{2i+3}], \ i = 0, \ldots, N - 1 \big\}, \end{aligned}$$

$$(2.6) \quad \begin{aligned} RS_1^0[t_0, t_{2N+1}] = \big\{ l \in C[t_0, t_{2N+1}]: l(x) \in \mathbf{P}_1 \text{ for } x \in [t_{2i}, t_{2i+2}], \\ i = 0, \ldots, N - 1; \ x \in [t_{2N}, t_{2N+1}] \big\} \end{aligned}$$

and

$$(2.7) \quad \begin{aligned} BS_1^0[t_0, t_{2N+1}] = \big\{ l \in C[t_0, t_{2N+1}]: l(x) \in \mathbf{P}_1 \text{ for } x \in [t_i, t_{i+1}], \\ i = 0, 1, \ldots, 2N \big\}. \end{aligned}$$

It is an easy task to see that

$$(2.8) \quad BS_1^0[t_0, t_{2N+1}] = LS_1^0[t_0, t_{2N+1}] + RS_1^0[t_0, t_{2N+1}].$$

We finally recall that the mth degree Bernstein polynomial of a function f in the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N - 1$, has the form [6]

$$B_i(x; f, m) = \frac{1}{h_i^m} \sum_{\nu=0}^{m} \binom{m}{\nu} f\left(x_i + \frac{\nu}{m} h_i\right)(x - x_i)^\nu (x_{i+1} - x)^{m-\nu}$$

and the following properties hold:

$$B_i^{(j)}(x; f, m)$$
$$(2.9) \qquad = \frac{1}{h_i^m} \frac{m!}{(m-j)!} \sum_{\nu=0}^{m-j} \binom{m-j}{\nu} \Delta^j f\left(x_i + \frac{\nu}{m} h_i\right)(x - x_i)^\nu (x_{i+1} - x)^{m-j-\nu},$$

$$(2.10) \qquad B_i^{(j)}(x_i; f, m) = \frac{1}{h_i^j} \frac{m!}{(m-j)!} \Delta^j f(x_i),$$

$$(2.11) \qquad B_i^{(j)}(x_{i+1}; f, m) = \frac{1}{h_i^j} \frac{m!}{(m-j)!} \nabla^j f(x_{i+1}),$$

where Δ^j (∇^j) is the jth forward (backward) difference with step h_i/m, $j = 1, 2, \dots, m$.

Now let $B: C[x_0, x_N] \to C[x_0, x_N]$ be the linear operator such that

$$(Bf)(x) = B_i(x; f, n_i) \quad \text{for } x \in [x_i, x_{i+1}], i = 0, \dots, N - 1.$$

It is simple to prove the following

LEMMA 1. *The restriction of B to $BS_1^0[t_0, t_{2N+1}]$ is an invertible map from $BS_1^0[t_0, t_{2N+1}]$ to $BS_n^k[x_0, x_N]$.*

Proof. Let $l \in BS_1^0[t_0, t_{2N+1}]$. Then we have from (2.8) that

$$l = \lambda + \mu, \quad \text{where } \lambda \in LS_1^0[t_0, t_{2N+1}] \text{ and } \mu \in RS_1^0[t_0, t_{2N+1}].$$

From (2.5), (2.6), (2.10) and (2.11) it is easy to see that

$$B_i^{(j)}(x_i; \lambda, n_i) = B_i^{(j)}(x_{i+1}; \mu, n_i) = 0, \qquad j = 2, \dots, k, i = 0, \dots, N - 1,$$
$$B_i^{(j)}(x_{i+1}; \lambda, n_i) = B_i^{(j)}(x_i; \mu, n_i) = 0, \qquad j = 2, \dots, n_i - k, i = 0, \dots, N - 1,$$

and $B_{i-1}^{(j)}(x_i; \lambda, n_{i-1}) = B_i^{(j)}(x_i; \lambda, n_i)$; $B_{i-1}^{(j)}(x_i; \mu, n_{i-1}) = B_i^{(j)}(x_i; \mu, n_i)$ for $j = 0, 1$, $i = 1, \dots, N - 1$. Hence, from (2.1), (2.2), $B\lambda \in LS_n^k[x_0, x_N]$, $B\mu \in RS_n^k[x_0, x_N]$, and definition (2.3) implies immediately that $Bl = B(\lambda + \mu) = B\lambda + B\mu \in BS_n^k[x_0, x_N]$.

On the other hand, let $s \in BS_n^k[x_0, x_N]$. Using (2.3), (2.1) and (2.2), we have that, for any $x \in [x_i, x_{i+1}]$, $i = 0, \dots, N - 1$, $s(x) = \phi(x) + \psi(x)$, where $\phi, \psi \in \mathbf{P}_{n_i}$ and

$$(2.12) \qquad \begin{aligned} \phi^{(j)}(x_i) &= \psi^{(j)}(x_{i+1}) = 0, \quad j = 2, \dots, k, \\ \phi^{(j)}(x_{i+1}) &= \psi^{(j)}(x_i) = 0, \quad j = 2, \dots, n_i - k. \end{aligned}$$

Since the two polynomials can be put in the form (see, e.g., [6, p. 48])

$$\phi(x) = \frac{1}{h_i^{n_i}} \sum_{\nu=0}^{n_i} \binom{n_i}{\nu} \lambda_\nu (x - x_i)^\nu (x_{i+1} - x)^{n_i - \nu},$$

$$\psi(x) = \frac{1}{h_i^{n_i}} \sum_{\nu=0}^{n_i} \binom{n_i}{\nu} \mu_\nu (x - x_i)^\nu (x_{i+1} - x)^{n_i - \nu},$$

we can write $\phi(x) = B_i(x; \lambda_i, n_i)$, $\psi(x) = B_i(x; \mu_i, n_i)$, where λ_i, μ_i are piecewise continuous linear functions such that

$$\lambda_i\left(x_i + \frac{\nu}{n_i}h_i\right) = \lambda_\nu; \quad \mu_i\left(x_i + \frac{\nu}{n_i}h_i\right) = \mu_\nu, \qquad \nu = 0, 1, \ldots, n_i.$$

Moreover, it follows from (2.12), (2.10) and (2.11) that they are made up of two linear segments which intersect each other respectively at t_{2i+1} and t_{2i+2} (see (2.4)). So the piecewise linear function $l_i = \lambda_i + \mu_i$ has two break points and, of course, $s(x) = B_i(x; l_i, n_i)$ for $x \in [x_i, x_{i+1}]$, $i = 0, \ldots, N - 1$. If we use again (2.10) and (2.11) for $j = 0, 1$, we have

$$l_{i-1}(x_i) = l_i(x_i), \quad \frac{1}{h_{i-1}}\nabla l_{i-1}(x_i) = \frac{1}{h_i}\Delta l_i(x_i), \qquad i = 1, \ldots, N,$$

and so $l \in BS_1^0[t_0, t_{2N+1}]$, where $l(x) = l_i(x)$, $x \in [x_i, x_{i+1}]$, $i = 0, \ldots, N - 1$. The obvious observation that $Bl = s$ then completes the proof.

The well-known shape-preserving properties of Bernstein polynomials ensure that, if $l \in BS_1^0[t_0, t_{2N+1}]$ is I, C, or IC, then the corresponding $s = Bl$ is also an I, C, or IC function.

Moreover, we can state the following

LEMMA 2. *Let $s \in BS_n^k[x_0, x_N]$ be C or IC. Then $l = B^{-1}s$ is also a C or IC linear spline. Further, if $k = n_i - k$, $i = 0, \ldots, N - 1$, and s is I, then also $l = B^{-1}s$ is increasing.*

Proof. Let $s \in BS_n^k[x_0, x_N]$ be any C or IC spline and let $l = B^{-1}s$, $l \in BS_1^0[t_0, t_{2N+1}]$. It follows from (2.7) that $\Delta l(x_i + \nu h_i/n_i) = 0$, $\nu = 0, 1, \ldots, n_i$, $\nu \neq k - 1$, $\nu \neq n_i - k - 1$, $i = 0, \ldots, N - 1$.

In the case $k = n_i - k$, using (2.9) we have

$$B_i^{(2)}(x; l_i, n_i) = \frac{1}{h_i^{n_i}} \frac{n_i!}{(n_i - 2)!} \binom{n_i - 2}{k - 1} \Delta^2 l\left(x_i + \frac{k - 1}{n_i}h_i\right)$$

and thus the convexity of s implies $\Delta^2 l(x) \geqslant 0$, $x \in [x_i, x_{i+1}]$, $i = 0, \ldots, N - 1$. For $k < n_i - k$ we have

$$B_i^{(2)}(x_i; l_i, n_i)$$
$$= [(x - x_i)(x_{i+1} - x_i)]^{k-1}\left[\lambda(x - x_i)^{n_i - 2k} + \mu(x_{i+1} - x)^{n_i - 2k}\right],$$

where

$$\lambda = \frac{1}{h_i^{n_i}} \frac{n_i!}{(n_i - 2)!} \binom{n_i - 2}{k - 1} \Delta^2 l\left(x_i + \frac{n_i - k - 1}{n_i}h_i\right),$$

$$\mu = \frac{1}{h_i^{n_i}} \frac{n_i!}{(n_i - 2)!} \binom{n_i - 2}{k - 1} \Delta^2 l\left(x_i + \frac{k - 1}{n_i}h_i\right).$$

If we suppose that l is not C, then λ or μ (or both) are negative. Suppose, without loss of generality, that $\mu < 0$. It is easy to see that the polynomial $[\lambda(x - x_i)^{n_i - 2k} + \mu(x_{i+1} - x)^{n_i - 2k}]$ has only a simple zero in (x_i, x_{i+1}), and, for some $x \in [x_i, x_{i+1}]$, $B_i^{(2)}(x; l_i, n_i) < 0$, which contradicts the hypotheses.

Moreover, $B_i^{(1)}(x_i; l_i, n_i) = n_i \Delta l(x_i)/h_i$, and this, together with the fact that any l which is C is also IC if $\Delta l(x_0) \geqslant 0$, will complete the proof for the IC case. Finally, if $k = n_i - k$, $i = 0, \ldots, n + 1$, then $\Delta l(x_i + \nu h_i/n_i) = \Delta l(x_i)$, $\nu = 0, \ldots, k - 1$, $\Delta l(x_i + \nu h_i/n_i) = \nabla l(x_{i+1})$, $\nu = k, \ldots, n - 1$, and for any I spline $s \in BS_n^k[x_0, x_N]$, $l = B^{-1}s$ is also increasing.

It is now clear that the existence of an I, C or IC interpolating spline, that is the existence of $s \in BS_n^k[x_0, x_N]$ such that $s(x_i) = y_i$, $i = 0, 1, \ldots, N$; $s^{(1)}(x) \geqslant 0$ or/and $s^{(2)}(x) \geqslant 0$, $x \in [x_0, x_N]$, is ensured if it is possible to construct an interpolating linear spline which is, moreover, I, C or IC. We note that, for any sequence d_i, $i = 0, 1, \ldots, N$, there is one and only one $l \in BS_1^0[t_0, t_{2N+1}]$, such that $l(x_i) = y_i$, $l^{(1)}(x_i) = d_i$, $i = 0, 1, \ldots, N$, where the values y_i are the given data, and d_i, $i = 0, 1, \ldots, N$, are free parameters.

So we can reformulate the problem into the following one: to compute the values d_i, $i = 0, 1, \ldots, N$, such that the corresponding $s \in BS_1^0[t_0, t_{2N+1}]$ is I, C, or IC. From definition (2.7) it is simple to obtain the following results, which are stated in the form of a lemma for notational convenience.

LEMMA 3. *Let* $l \in BS_1^0[t_0, t_{2N+1}]$ *be such that* $l(x_i) = y_i$, $l^{(1)}(x_i) = d_i$, $i = 0, 1, \ldots, N$. *Then* l *is I in* $[x_i, x_{i+1}]$, $i = 0, \ldots, N - 1$, *if, and only if,* $(d_i, d_{i+1}) \in ID_i$ *where*

(2.13.a) $\quad ID_i = \left\{ (u, v) \in \mathbf{R}^2 : u \geqslant 0, v \geqslant 0, v \leqslant -u + \dfrac{n_i}{k} \Delta_i \right\}$ *if* $k < n_i - k$,

(2.13.b) $\quad ID_i = \left\{ (u, v) \in \mathbf{R}^2 : u \geqslant 0, v \geqslant 0, v = -u + 2\Delta_i \right\}$ *if* $k = n_i - k$;

is C if, and only if, $(d_i, d_{i+1}) \in CD_i$, *where*

(2.14)
$$CD_i = \left\{ (u, v) \in \mathbf{R}^2 : u \leqslant \Delta_i, v \geqslant -\frac{k}{n_i - k} u + \frac{n_i}{n_i - k} \Delta_i, \right.$$
$$\left. v \leqslant -\frac{n_i - k}{k} u + \frac{n_i}{k} \Delta_i \right\};$$

is IC if, and only if, $(d_i, d_{i+1}) \in ICD_i$, *where*

(2.15) $\qquad\qquad\qquad ICD_i = \left\{ (u, v) \in CD_i : u \geqslant 0 \right\}.$

The nonempty convex sets ID_i, CD_i, ICD_i *are shown in Figure* 1.

We see at this point that the problem of I, C, or IC spline interpolation is reduced to finding a sequence of slopes d_i, $i = 0, 1, \ldots, N$, such that every ordered couple (d_i, d_{i+1}), $i = 0, \ldots, N - 1$, belongs to ID_i, CD_i or ICD_i, respectively.

A more general formulation of this problem is given below. Let G_i, $i = 0, 1, \ldots, N$, be arbitrary nonempty sets, and let the domains $D_i \subseteq G_i \times G_{i+1}$, $D_i \neq \varnothing$, $i = 0, \ldots, N - 1$, be given.

Problem P. Do there exist elements $g_i \in G_i$, $i = 0, 1, \ldots, N$, such that $(g_i, g_{i+1}) \in D_i$, $i = 0, \ldots, N - 1$?

Defining the projective operators $\Pi_{i,i+1}^j : G_i \times G_{i+1} \to G_j$, $j = i, i + 1$, $i = 0, \ldots, N - 1$, and the sets $B_i \subseteq G_i$, $i = 0, 1, \ldots, N$, where

(2.16) $\qquad\qquad B_i = \Pi_{i,i+1}^i(D_i)$, $\quad i = 0, \ldots, N - 1$; $\qquad B_N = G_N$,

we introduce the following algorithm and lemma and give an answer to Problem P.
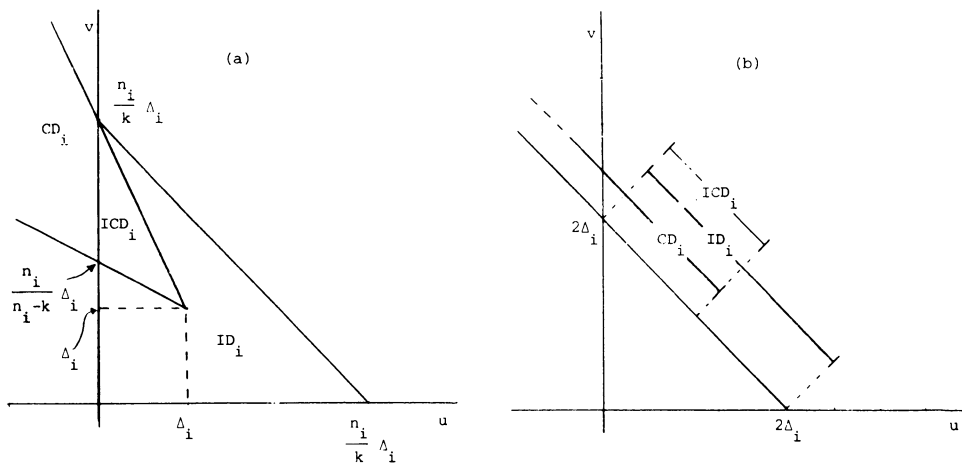
FIGURE 1

The domains ID_i, CD_i, ICD_i for (a) $k < n_i - k$, (b) $k = n_i - k$.

ALGORITHM A.
1. Set $A_0 = B_0$, $J = N$.
2. For $i = 1, \ldots, N$
    1. $P = \Pi^i_{i-1,i}(\{A_{i-1} \times G_i\} \cap D_{i-1})$;
    2. $A_i = B_i \cap P$;
    3. If $A_i = \varnothing$, set $J = i$ and stop. Otherwise continue.
3. Stop.

LEMMA 4. *Let J be computed as in Algorithm* A.
(a) *If $J = N$, then there are solutions to Problem* P.
(b) *If $J = N$ and $g_i \in G_i$, $i = 0, 1, \ldots, N$, are such that $g_j \notin A_j$ for some j, $1 \leqslant j \leqslant N$, then the sequence g_i, $i = 0, 1, \ldots, N$, is not a solution to Problem* P.
(c) *If $J = j \neq N$, then there are no solutions to Problem* P.

Proof. Let $J = N$ and $g_N \in A_N$. It follows from steps 2.2, 2.1 in the algorithm that $g_N \in \Pi^N_{N-1,N}(\{A_{N-1} \times G_N\} \cap D_{N-1})$ and so, from (2.16), there exists $g_{N-1} \in G_{N-1}$ such that $(g_{N-1}, g_N) \in D_{N-1}$ and $g_{N-1} \in A_{N-1}$.

Since the above arguments hold for any i, $i = N, \ldots, 1$, the proof of statement (a) follows from the induction principle.

Now let $J = N$, $g_i \in G_i$, $i = 0, 1, \ldots, N$, and $g_j \notin A_j$, $1 \leqslant j \leqslant N$. We note that steps 2.1 and 2.2 are equivalent to setting

(2.17) $A_i = \Pi^i_{i-1,i}(\{A_{i-1} \times B_i\} \cap D_{i-1})$,

and so $(g_{j-1}, g_j) \notin \{A_{j-1} \times B_j\} \cap D_{j-1}$. Hence, if $(g_{j-1}, g_j) \notin D_{j-1}$, the proof of statement (b) is complete. Suppose that $(g_{j-1}, g_j) \in D_{j-1}$ and, consequently, $(g_{j-1}, g_j) \notin \{A_{j-1} \times B_j\}$. If $g_j \notin B_j = \Pi^j_{j,j+1}(D_j)$, we have $(g_j, g_{j+1}) \notin D_j$ and again (b) is proved. If $g_{j-1} \notin A_{j-1}$, we can repeat the above arguments for the index $j - 1$. We note that $A_0 = B_0 = \Pi^0_{0,1}(D_0)$ and, from $g_0 \notin A_0$, it follows immediately that $(g_0, g_1) \notin D_0$. This completes the proof of statement (b).
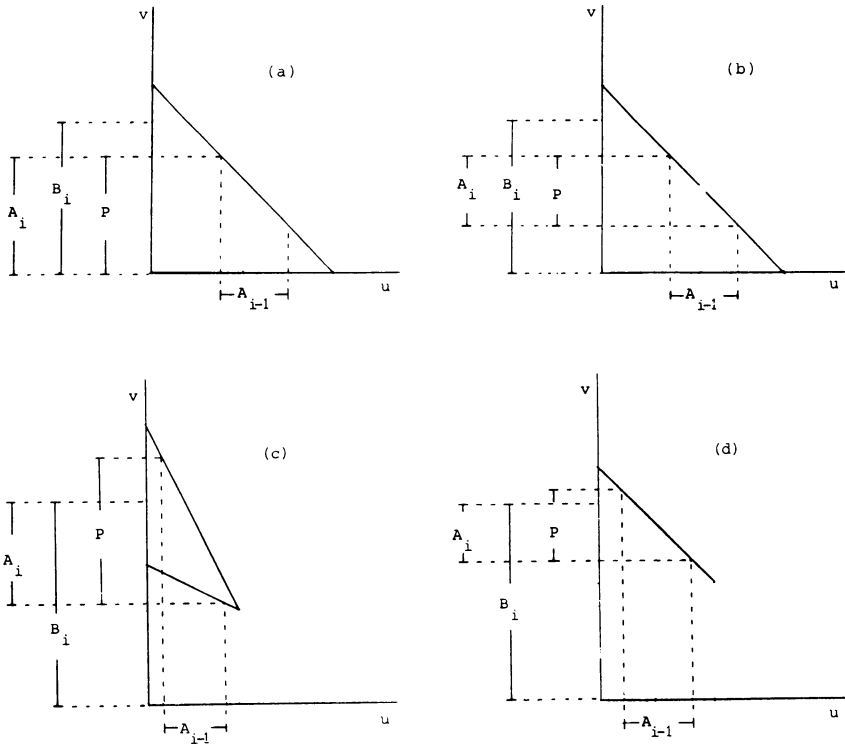
FIGURE 2

Geometric interpretation of steps 2.1, 2.2 *of Algorithm* A.
(a) *I data*, $k < n_{i-1} - k$; (b) *I data*, $k = n_{i-1} - k$; (c) *IC data*,
$k < n_{i-1} - k$; (d) *IC data*, $k = n_{i-1} - k$.

Finally let $J = j \neq N$. This means that $A_j = \varnothing$, $A_{j-1} \neq \varnothing$ and, from (2.17), $\{A_{j-1} \times B_j\} \cap D_{j-1} = \varnothing$. Since this implies that, for any sequence g_i, $i = 0, 1, \ldots, N$, $(g_{j-1}, g_j) \notin \{A_{j-1} \times B_j\} \cap D_{j-1}$, the proof of statement (c) is similar to the previous one.

Now we can apply Lemma 4 to the sets ID_i, CD_i, ICD_i given in Lemma 3. Before doing so, we point out that, in this case, $G_i \equiv \mathbf{R}$, $i = 0, 1, \ldots, N$, $B_i \equiv [0, n_i \Delta_i / k]$ or $B_i \equiv (-\infty, \Delta_i]$ or $B_i \equiv [0, \Delta_i]$, $i = 0, \ldots, N - 1$, for I, C or IC data, respectively. A geometric interpretation of steps 2.1 and 2.2 of Algorithm A, for I or IC data is given in Figure 2. We note that C and IC cases have an identical representation.

Setting $a_i^{(1)} = \inf\{u \in A_i\}$, $a_i^{(2)} = \sup\{u \in A_i\}$, $i = 0, 1, \ldots, N$, and using (2.13), (2.14), (2.15), it is possible to rewrite the general Algorithm A in more detailed forms, for I, C or IC data, respectively.

ALGORITHM A/I.
1. Set $a_0^{(1)} = 0$, $a_0^{(2)} = n_0 \Delta_0 / k$; $J = N$; $n_N = k$; $\Delta_N = +\infty$.
2. For $i = 1, \ldots, N$
    1. $a_i^{(1)} = 0$ if $k < n_{i-1} - k$, or $a_i^{(1)} = -a_{i-1}^{(2)} + 2\Delta_{i-1}$ if $k = n_{i-1} - k$;
       $a_i^{(2)} = \min\{n_i \Delta_i / k; -a_{i-1}^{(1)} + n_{i-1} \Delta_{i-1} / k\}$.
    2. If $a_i^{(2)} < a_i^{(1)}$ set $J = i$ and stop. Otherwise continue.
3. Stop.

ALGORITHM A/C.

1. Set $a_0^{(1)} = -\infty$, $a_0^{(2)} = \Delta_0$; $J = N$; $\Delta_N = +\infty$.

2. For $i = 1, \ldots, N$

   1. $a_i^{(1)} = -(k/(n_{i-1} - k))a_{i-1}^{(2)} + n_{i-1}\Delta_{i-1}/(n_{i-1} - k)$;
      $a_i^{(2)} = \min\{\Delta_i; -((n_{i-1} - k)/k)a_{i-1}^{(1)} + n_{i-1}\Delta_{i-1}/k\}$.

   2. If $a_i^{(2)} < a_i^{(1)}$ set $J = i$ and stop. Otherwise continue.

3. Stop.

ALGORITHM A/IC.

1. Set $a_0^{(1)} = 0$, $a_0^{(2)} = \Delta_0$; $J = N$; $\Delta_N = +\infty$.

2. For $i = 1, \ldots, N$

   1. $a_i^{(1)} = -(k/(n_{i-1} - k))a_{i-1}^{(2)} + n_{i-1}\Delta_{i-1}/(n_{i-1} - k)$;
      $a_i^{(2)} = \min\{\Delta_i; -((n_{i-1} - k)/k)a_{i-1}^{(1)} + n_{i-1}\Delta_{i-1}/k\}$.

   2. If $a_i^{(2)} < a_i^{(1)}$ set $J = i$ and stop. Otherwise continue.

3. Stop.

Linking together Lemmas 1, 2 and 4, we can state the following

THEOREM 1. *Let I, C or IC data* (x_i, y_i), $i = 0, 1, \ldots, N$, *be given, and let J be computed by Algorithm A/I, A/C or A/IC, respectively. There exist I interpolating splines* $s \in BS_n^k[x_0, x_N]$ *if* $J = N$. *If* $k = n_i - k$, $i = 0, \ldots, N - 1$, *there exist I interpolating splines s only if* $J = N$. *There exist C or IC interpolating splines* $s \in BS_n^k[x_0, x_N]$ *if, and only if,* $J = N$.

We conclude this section by noting that, if $k < n_i - k$, $i = 0, \ldots, N - 1$, and the data are increasing, then Algorithm A/I will yield $a_i^{(1)} < a_i^{(2)}$, $i = 0, 1, \ldots, N$. In other words, we obtain the well-known fact (see e.g., [5]) that there always exists an I interpolating spline $s \in S_n^k[x_0, x_N]$ whenever $k < n - k$.

3. Degree-Dependent Existence Conditions. In the previous section we noted that it is always possible to find an I interpolating spline $s \in S_n^k[x_0, x_N]$ when $k < n - k$. On the other hand, it is shown in [9] that, if C interpolation is desired, then the data can force n to be very large, and that this fact is inherent in the nature of C spline interpolation. These considerations suggest to investigate the relationship between the degrees n_i of the various polynomial pieces of the spline, and the "roughness" of data in case of C or IC interpolation. The results, which are easily obtained from Algorithms A/C and A/IC, are collected in two theorems.

THEOREM 2. *Let* $\Delta_{i+1} > \Delta_i$, $i = 0, \ldots, N - 2$, *and let the sequence* n_i, $i = 0, \ldots, N - 1$, *be such that*

(3.1)
$$n_0 \geqslant 2k; \qquad n_{N-1} \geqslant 2k,$$
$$n_i \geqslant \max\left\{2k, k\frac{\Delta_{i+1} - \Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right\}, \qquad i = 1, \ldots, N - 2.$$

Then there exist C interpolating splines $s \in BS_n^k[x_0, x_N]$. *If* $\Delta_0 > 0$ *and*

(3.2)
$$n_0 \geqslant \max\left\{2k, k\frac{\Delta_1}{\Delta_0}\right\},$$

then there exist IC interpolating splines $s \in BS_n^k[x_0, x_N]$.

Proof. If we execute step 2.1 of Algorithm A/C for $i = 1$, we obtain $a_1^{(1)} = \Delta_0$, $a_1^{(2)} = \Delta_1$ and so $a_1^{(1)} < a_1^{(2)}$. Now suppose that, for some i, $2 \leqslant i \leqslant N - 1$, $a_{i-1}^{(1)} = \Delta_{i-2}$, $a_{i-1}^{(2)} = \Delta_{i-1}$. Step 2.1 gives

$$a_i^{(1)} = \Delta_{i-1}, \qquad a_i^{(2)} = \min\left\{\Delta_i; \ -\frac{n_{i-1} - k}{k}\Delta_{i-2} + \frac{n_{i-1}}{k}\Delta_{i-1}\right\},$$

and, from (3.1), we have again $a_i^{(1)} = \Delta_{i-1}$, $a_i^{(2)} = \Delta_i$; $a_i^{(1)} < a_i^{(2)}$. Step 2.1 for $i = N$ gives

$$a_N^{(1)} = \Delta_{N-1}, \qquad a_N^{(2)} = -\frac{n_{N-1} - k}{k}\Delta_{N-2} + \frac{n_{N-1}}{k}\Delta_{N-1}$$

and so $a_N^{(1)} < a_N^{(2)}$.

The proof of the first statement is completed using Theorem 1. Executing step 2.1 of Algorithm A/IC for $i = 1$, we obtain $a_1^{(1)} = \Delta_0$ and $a_1^{(2)} = \min\{\Delta_1; \ n_0\Delta_0/k\}$. Since (3.2) implies $\Delta_1 < n_0\Delta_0/k$, the proof of the second statement is identical to the previous one.

The next step is to give an analogue of Theorem 2 for nonstrictly C or IC data. We need an intermediate result given in the lemma below.

LEMMA 5. *Let $\Delta_{i+1} > \Delta_i$, $i = 0, \ldots, N - 2$, and let the sequence n_i, $i = 0, \ldots, N - 1$, be such that*

$$n_0 \geqslant 2k, \qquad n_{N-1} \geqslant 2k,$$

$$n_1 \geqslant \max\left\{2k; \ k\frac{\Delta_2 - \Delta_0}{\Delta_1 - \Delta_0}\right\}, \qquad n_1 > k\frac{\Delta_2 - \Delta_0}{\Delta_2 - \Delta_1},$$

(3.3)
$$n_2 \geqslant \max\left\{2k; \ \frac{k^2(\Delta_0 - \Delta_3) + kn_1(\Delta_3 - \Delta_1)}{(n_1 - k)\Delta_2 + k\Delta_0 - n_1\Delta_1}\right\},$$

$$n_i \geqslant \max\left\{2k; \ k\frac{\Delta_{i+1} - \Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right\}, \qquad i = 3, \ldots, N - 2.$$

Then there exist C interpolating splines $s \in BS_n^k[x_0, x_N]$ such that

(3.4)
$$d_0 = s^{(1)}(x_0) = \Delta_0; \qquad d_N = s^{(1)}(x_N) = \Delta_{N-1}.$$

Proof. In this case we must apply directly Algorithm A with $G_i \equiv \mathbf{R}$, $i = 0, 1, \ldots, N$;

$$D_i \equiv CD_i, \qquad i = 1, \ldots, N - 2 \ (\text{see } (2.14));$$

$$D_0 = CD_0 \cap \{(u, v) \in \mathbf{R}^2 : u = \Delta_0\} = \{(\Delta_0, \Delta_0)\}$$

and

$$D_{N-1} = CD_{N-1} \cap \{(u, v) \in \mathbf{R}^2 : v = \Delta_{N-1}\} = \{(\Delta_{N-1}, \Delta_{N-1})\}.$$

The boundary conditions (3.4) are implicitly introduced by the above definitions, and so it will be sufficient to prove that Algorithm A will yield $J = N$. We note also that, in this case, Algorithm A is a slightly modified version of Algorithm A/C.

Steps 2.1 and 2.2 for $i = 1$ give $a_1^{(1)} = a_1^{(2)} = \Delta_0$ and, for $i = 2$, we have (see A/C; step 2.1)

$$a_2^{(1)} = -\frac{k}{n_1 - k}\Delta_0 + \frac{n_1}{n_1 - k}\Delta_1; \qquad a_2^{(2)} = \min\left\{\Delta_2; \ -\frac{n_1}{n_1 - k}\Delta_0 + \frac{n_1}{k}\Delta_1\right\}$$

and, by (3.3), $a_2^{(2)} = \Delta_2$, $a_2^{(1)} < a_2^{(2)}$. For $i = 3$ steps 2.1, 2.2 give

$$a_3^{(1)} = \Delta_2; \qquad a_3^{(2)} = \min\left\{\Delta_3; \; -\frac{n_2}{n_2 - k}a_2^{(1)} + \frac{n_2}{k}\Delta_2\right\},$$

and (3.3) implies $a_3^{(2)} = \Delta_3$ and consequently $a_3^{(1)} < a_3^{(2)}$. Now it is not difficult to proceed as in Lemma 5 and show that $\Delta_{i-1} = a_i^{(1)} < a_i^{(2)} = \Delta_i$, $i = 4, \ldots, N - 2$. Finally, steps 2.1, 2.2 for $i = N - 1$, $i = N$ and the definition of D_{N-1}, give $a_{N-1}^{(1)} = a_{N-1}^{(2)} = \Delta_{N-1}$ and $a_N^{(1)} = a_N^{(2)} = \Delta_{N-1}$. Theorem 1 then completes the proof.

We note that, for IC data, $\Delta_0 \geqslant 0$, and the boundary conditions (3.4) imply that the interpolating spline $s \in BS_n^k[x_0, x_N]$ is also IC. For nonstrictly C or IC data, it is worthwhile to note that if

(3.5) $\Delta_{i-2} = \Delta_{i-1} \neq \Delta_i = \Delta_{i+1}$ for some i, $2 \leqslant i \leqslant N - 2$,

it is impossible to find differentiable interpolating C or IC functions. When the data have no "contiguous linear pieces", an extension of Theorem 2 can be given. The idea is to divide the data (x_i, y_i), $i = 0, 1, \ldots, N$, into more subsets of linear or strictly C data, and use Lemma 5 to guarantee the existence of more C interpolating splines, sharing the same boundary conditions. A formal description is given in Theorem 3, whose proof is a consequence of Lemma 5.

THEOREM 3. *Suppose the C or IC data do not satisfy* (3.5). *Let* $L = \{i \in \mathbf{N}$, $0 \leqslant i \leqslant N - 2: \Delta_i = \Delta_{i+1}\}$ *and* $\bar{L} = \{i \in \mathbf{N}, 0 \leqslant i \leqslant N - 1: i \notin L\}$, *and let the sequence* n_i, $i = 0, \ldots, N - 1$, *be such that*

$$n_i \geqslant 2k \text{ for } i = 0, i = N - 1 \text{ or } i \in L;$$

$$n_i \geqslant \max\left\{2k; \; k\frac{\Delta_{i+1} - \Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right\}, \qquad n_i > k\frac{\Delta_{i+1} - \Delta_{i-1}}{\Delta_{i+1} - \Delta_i},$$

$$n_{i+1} \geqslant \max\left\{2k; \; \frac{k^2(\Delta_{i-1} - \Delta_{i+2}) + kn_i(\Delta_{i+2} - \Delta_i)}{(n_i - k)\Delta_{i+1} + k\Delta_{i-1} - n_i\Delta_i}\right\} \quad and$$

$$n_{i-1} \geqslant 2k \quad for \; i - 2 \in L \text{ and } i - 1 \in \bar{L};$$

$$n_i \geqslant \max\left\{2k; \; k\frac{\Delta_{i+1} - \Delta_{i-1}}{\Delta_i - \Delta_{i-1}}\right\} \quad otherwise.$$

Then there exist C or IC interpolating splines $s \in BS_n^k[x_0, x_N]$.

4. Some Remarks. As previously mentioned in the introduction, the well-known results of [9], [7] can be deduced from those presented here. In this paper all the lemmas and theorems are stated for k fixed. However, they are still valid if we introduce, together with the polynomial degrees n_i, $i = 0, \ldots, N - 1$, the sequence k_i, $i = 0, \ldots, N - 1$. This means (see (2.4)) that $t_{2i+1} = x_i + k_i h_i/n_i$, $t_{2i+2} = x_i + (n_i - k_i)h_i/n_i$, $i = 0, \ldots, N - 1$, and that $n = \max\{n_i, \; i = 0, \ldots, N - 1\}$, $k = \min\{k_i, \; i = 0, \ldots, N - 1\}$. Moreover, if we are interested in the construction of linear splines with only one knot in (x_i, x_{i+1}), $i = 0, \ldots, N - 1$, the restriction $k_i \leqslant n_i - k_i$ is no longer useful and it is sufficient to require $k_i < n_i$, $i = 0, \ldots, N - 1$ with $k = \min\{k_i; \; n_i - k_i; \; i = 0, \ldots, N - 1\}$. So we placed ourselves in the hypotheses of Passow and Roulier, and Theorem 2 of [9] can be deduced by a slightly modified version of Theorem 1. Setting $\alpha_i = k_i/n_i$, $i = 0, \ldots, N - 1$, and,

as in the previous sections, $d_i = l^{(1)}(x_i) = s^{(1)}(x_i)$, $i = 0, 1, \ldots, N$, if the interpolating linear spline l has a knot at $x_i + \alpha_i h_i$, then

$$y_i + d_i \alpha_i h_i = y_i - d_{i+1}(1 - \alpha_i) h_i.$$

If we require that l is IC then we obtain

$$(d_i, d_{i+1}) \in \left\{ (u, v) \in \mathbf{R}^2 \colon 0 \leqslant u \leqslant \Delta_i; \; v = -\frac{\alpha_i}{1 - \alpha_i} u + \frac{\Delta_i}{1 - \alpha_i} \right\},$$

$$i = 0, \ldots, N - 1,$$

and Algorithm A and Lemma 4 applied to this problem are equivalent, respectively, to Alpha-Algorithm and Theorem 3.2 of [7].

If we are interested in the problem of finding a C or IC interpolating spline $s \in BS_n^k[x_0, x_N]$ for $n_i > 2$, $i = 0, \ldots, N - 1$, and k fixed, then the use of splines with two knots (instead of only one) in (x_i, x_{i+1}), not only enlarges the family of data sets which allow C or IC spline interpolation, but has another practical advantage. In fact, a change of the slope d_i at x_i has a local effect in $[x_{i-1}, x_{i+1}]$ (since it is absorbed by the contiguous linear segments in $[t_{2i-1}, t_{2i}]$ and $[t_{2i+1}, t_{2i+2}]$) while in the other case the effect is extended to the whole interval $[x_0, x_N]$. From a practical point of view, this implies that, if Algorithm A/C or A/IC gives $J = j \neq N$, we can apply it again from j to N, construct two (or more) splines interpolating the sets (x_i, y_i), $i = 0, \ldots, j$, and (x_i, y_i), $i = j, \ldots, N$, and then join together their derivatives at x_j. Hence the resulting spline is C or IC everywhere except for, at most, $x \in [x_{j-1}, x_{j+1}]$, but can still be I.

With respect to Algorithm A and Lemma 4, it is worthwhile to note that they are stated in a general form, and seem useful for the construction of "constrained separable splines", that is, splines which can be locally defined between every pair of knots, and are subject to some restrictions. For instance, they are used in [4] for the construction of piecewise quadratic histosplines. We remark that Algorithm A does not require that the data are all I or C or IC, and, with a suitable definition of the sets D_i, $i = 0, \ldots, N - 1$ (see Problem P), the resulting code can produce shape-preserving spline interpolation without preliminary subdivisions of the points. Moreover, auxiliary Hermite interpolation conditions can be added at some knots (see the proof of Lemma 5), as well as any other local constraint. From a theoretical point of view, interpolating splines could be introduced without the use of Bernstein polynomials. For example, an IC interpolating $s \in S_5^1[x_0, x_N]$ is determined requiring $s^{(1)}(x_i) = d_i$, $s^{(2)}(x_i) = e_i$, $i = 0, 1, \ldots, N$, and the problem is reduced to the following: Determine the sequences d_i, e_i, $i = 0, 1, \ldots, N$, such that $(d_i, e_i, d_{i+1}, e_{i+1}) \in D_i$, $i = 0, \ldots, N - 1$, where $D_i \subset \mathbf{R}^4$ is the IC region for polynomials $p \in \mathbf{P}_5$. Unfortunately, it is practically impossible to implement steps 2.1 and 2.2 of Algorithm A for general subsets of $\mathbf{R}^n$ and, even if it were, the computational cost would be very high. We conclude this section by noting that, since $S_3^1[x_0, x_N] \equiv BS_3^1[x_0, x_N]$, Theorems (2.2) and (2.3) of [2] can be obtained from Theorem 1.

Dipartimento di Energetica
Università degli Studi di Firenze
Via di S. Marta, 3
I 50139 Firenze, Italy

1. P. COSTANTINI, "An algorithm for computing shape-preserving interpolating splines of arbitrary degree," *J. Comput. Appl. Math.* (Submitted.)

2. P. COSTANTINI & R. MORANDI, "Monotone and convex cubic spline interpolation," *Calcolo*, v. 21, 1984, pp. 281–294.

3. P. COSTANTINI & R. MORANDI, "An algorithm for computing shape-preserving cubic spline interpolation to data," *Calcolo*, v. 21, 1984, pp. 295–305.

4. P. COSTANTINI & R. MORANDI, "Piecewise monotone quadratic histosplines." (Preprint.)

5. C. DE BOOR & B. SWARTZ, "Piecewise monotone interpolation," *J. Approx. Theory*, v. 21, 1977, pp. 411–416.

6. G. G. LORENTZ, *Bernstein Polynomials*, University of Toronto Press, Toronto, 1953.

7. D. F. MCALLISTER, E. PASSOW & J. A. ROULIER, "Algorithms for computing shape-preserving spline interpolation to data," *Math. Comp.*, v. 31, 1977, pp. 717–725.

8. E. NEUMAN, *Shape-Preserving Interpolation by Polynomial Splines*, Report no. N112, Institute of Computer Science, Wrocław University, 1982.

9. E. PASSOW & J. A. ROULIER, "Monotone and convex spline interpolation," *SIAM J. Numer. Anal.*, v. 14, 1977, pp. 904–909.